

Problem K. Shuffle

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 256 mebibytes

Byteasar has learnt a fabulous recursive method of shuffling a deck of cards. This algorithm can be described as follows:

- In order to shuffle two cards, swap them.
- In order to shuffle 2^k cards ($k \geq 2$), split them into two equal parts – an upper one and a lower one (so that each of them has 2^{k-1} cards). Shuffle each of them recursively and then put the lower half on the top of the upper half.

Byteasar has a deck of 2^n cards. Each of them has a number written on it. Byteasar now shuffles the deck, running the procedure described above exactly t times. As it might take a great amount of time, he would like to know the final order of the cards beforehand.

Input

The first line of the input contains two integers n, t ($1 \leq n \leq 20, 1 \leq t \leq 10^9$). The second line contains 2^n integers a_1, \dots, a_{2^n} ($1 \leq a_i \leq 10^9$); a_i is the number written on the i -th topmost card in the deck.

Output

In the first and only line of output print 2^n integers – the numbers written on the cards of Byteasar's deck after t shuffles. Print the numbers in the order from the topmost to the bottommost card.

Example

standard input	standard output
2 1 2 4 1 5	5 1 4 2